

PCTWORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G06F 13/00, H04L 12/46	A2	(11) International Publication Number: WO 99/00737 (43) International Publication Date: 7 January 1999 (07.01.99)
(21) International Application Number: PCT/US98/13203 (22) International Filing Date: 24 June 1998 (24.06.98) (30) Priority Data: 08/885,000 30 June 1997 (30.06.97) US (71) Applicant: SUN MICROSYSTEMS, INC. [US/US]; 901 San Antonio Road, Palo Alto, CA 94303 (US). (72) Inventors: MULLER, Shimon; Apartment D, 983 La Mesa Terrace, Sunnyvale, CA 94086 (US). YEUNG, Louise; 110 Rogers Avenue, San Carlos, CA 94070 (US). HENDEL, Ariel; 7537 Newcastle Drive, Cupertino, CA 95014 (US). (74) Agents: HYMAN, Eric, S. et al.; Blakely, Sokoloff, Taylor & Zafman, 7th floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025-1026 (US).		(81) Designated States: JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>Without international search report and to be republished upon receipt of that report.</i>
(54) Title: MECHANISM FOR PACKET FIELD REPLACEMENT IN A MULTI-LAYERED SWITCHED NETWORK ELEMENT (57) Abstract A system and method for updating packet headers using hardware that maintains the high performance of the network element. In one embodiment, the system includes an input port process (IPP) that buffers the input packet received and forwards header information to the search engine. The search engine searches a database maintained on the switch element to determine the type of the packet. In one embodiment, the type may indicate whether the packet can be routed in hardware. In another embodiment, the type may indicate whether the packet supports VLANs. The search engine sends the packet type information to the IPP along with the destination address (DA) to be updated if the packet is to be routed, or a VLAN tag if the packet has been identified to be forwarded to a particular VLAN. The IPP, during transmission of the packet to a packet memory selectively replaces the corresponding fields, e.g., DA field or VLAN tag field; the modified packet is stored in the packet memory. Associated with the packet memory are control fields containing control field information conveyed to the packet memory by the IPP. An output port process (OPP) reads the modified input packet and the control field information and selectively performs additional modifications to the modified input packet and issue control signals to the output interface (i.e., MAC). The MAC, based upon the control signals, replaces the source address field with the address of the MAC and generates a CRC that is appended to the end of the packet.		

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav	TM	Turkmenistan
BF	Burkina Faso	GR	Greece		Republic of Macedonia	TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's	NZ	New Zealand		
CM	Cameroon		Republic of Korea	PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakhstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

MECHANISM FOR PACKET FIELD REPLACEMENT IN A MULTI-LAYERED SWITCHED
NETWORK ELEMENT

FIELD OF THE INVENTION

5 The system and method of the present invention relates to the area of packet field replacement in a packet switched network; more particularly the present invention relates to the hardware implementation of packet field replacement of packet header fields of a packet within a switch coupled to a network.

10

ART BACKGROUND

Local area networks (LANs) have become quite sophisticated in architecture. Originally, LANs were thought of a single wire connecting a few
15 computers. Today LANs are implemented in complicated configurations to enhance functionality and flexibility. In such a network, packets are transmitted from source device to destination device; in more expansive networks, this packet can travel through one or more switches and/or routers. Standards have been set to define the packet structure and layers of
20 functionality and sophistication of a network. For example, the TCP/IP protocol stack defines four distinct multiple layers, e.g. the physical layer (layer 1), data link layer (layer 2), network layer (layer 3), transport layer (layer 4). A network device may be capable of support in one or more of the layers and refer to particular fields of the header accordingly.

25 Today, typical LANs utilize a combination of Layer 2 (data link layer) and Layer 3 (network layer) network devices. In order to meet the ever increasing performance demands from the network, functionality that has

been traditionally performed in software and/or in separate layer 2 and layer 3 devices have migrated into one multi-layer device or switch that implements the performance critical functions in hardware.

One performance critical function is routing. Software that
5 implements routing can impact performance. Therefore, it is desirable to implement the routing in faster hardware. However, routing requires certain header fields of an incoming packet to be modified prior to output from the device. Although perhaps straightforward to perform in software, in a hardware implementation, it is critical to minimize additional hardware
10 while not compromising performance.

Recently, the concept of Virtual Local Area Networks (VLANs) was introduced to Layer 2. The Layer 2 header has been modified to add bits that provide VLAN capability. VLANs enable the logical partitioning of network nodes independent of physically partitioning or arrangement in the network
15 topology. Based upon the state of the packet, VLAN bits, e.g., VLAN tags may also need to be modified. Although software-based techniques are usable, it is desirable to provide an efficient hardware approach.

SUMMARY OF THE INVENTION

A system and method for updating packet headers using hardware that provides minimal impact on performance of the network element. In one embodiment, the system includes an input port process (IPP) that buffers the input packet received and forwards header information to the search engine. The search engine searches a database maintained on the switch element to determine the type of the packet. In one embodiment, the type may indicate whether the packet can be routed in hardware. In another embodiment, the type may indicate whether the packet supports VLANs. The search engine sends the packet type information to the IPP along with the destination address (DA) to be updated if the packet is to be routed, or a VLAN tag if the packet has been identified to be forwarded to a particular VLAN. The IPP, during transmission of the packet to a packet memory selectively replaces the corresponding fields, e.g., DA field or VLAN tag field; the modified packet is stored in the packet memory. Associated with the packet memory are control fields containing control field information conveyed to the packet memory by the IPP. In one embodiment, the control field information consists of a flag to indicate that the source address needs replacement. In another embodiment, the control field information consists of flags to indicate whether the packet came in tagged, the packet came in tagged but the tag was modified or that the packet is not to be tagged.

An output port process (OPP) reads the modified input packet and the control field information, selectively performs additional modifications to the modified input packet and issues control signals to the output interface (i.e., MAC). In one embodiment, the OPP strips the last 4 bytes of the packet corresponding to the CRC and asserts control signals to the MAC to append a

CRC and replace the source address. In another embodiment, depending upon the state of the three control fields, the OPP removes the VLAN tag field in the packet, strips the last 4 bytes of the packet corresponding to the CRC and issues a control signal to the MAC to append a CRC. The MAC,
5 based upon the control signals received from the OPP, replaces the source address field with its own MAC address and generates a CRC that is appended to the end of the packet.

BRIEF DESCRIPTION OF THE DRAWINGS

The objects, features and advantages of the present invention will be apparent to one skilled in the art in which:

5

Figure 1 is a simplified block diagram of a high speed switch which operates in accordance with the teachings of the present invention.

Figure 2 is a simplified block diagram of a high speed switch element
10 which operates in accordance with the teachings of the present invention.

Figure 3 is a simplified block diagram of one embodiment of the system of the present invention.

15 Figure 4a illustrates a packet format and Figure 4b illustrates a packet format with VLAN support.

Figure 5 is a simplified flow diagram of one embodiment of the method of the present invention.

20

Figures 6 is a simplified flow diagram of one embodiment of the method for packet field modification in accordance with the teachings of the present invention.

25 Figures 7a and 7b are simplified flow diagrams of another embodiment of the method for packet field modification in accordance with the teachings of the present invention.

DETAILED DESCRIPTION

In the following description, for purposes of explanation, numerous details are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that these specific details are not required in order to practice the present invention. In other instances, well known electrical structures and circuits are shown in block diagram form in order not to obscure the present invention unnecessarily.

10 An overview of one embodiment of a network element that operates in accordance with the teachings of the present invention is illustrated in Figure 1. The network element is used to interconnect a number of nodes and end-stations in a variety of different ways. In particular, an application of the multi-layer distributed network element (MLDNE) would be to route
15 packets according to predefined routing protocols over a homogenous data link layer such as the IEEE 802.3 standard, also known as the Ethernet. Other routing protocols can also be used.

 The MLDNE's distributed architecture can be configured to route message traffic in accordance with a number of known or future routing
20 algorithms. In a preferred embodiment, the MLDNE is configured to handle message traffic using the Internet suite of protocols, and more specifically the Transmission Control Protocol (TCP) and the Internet Protocol (IP) over the Ethernet LAN standard and medium access control (MAC) data link layer. The TCP is also referred to here as a Layer 4 protocol, while the IP is referred
25 to repeatedly as a Layer 3 protocol.

 In one embodiment of the MLDNE, a network element is configured to implement packet routing functions in a distributed manner, i.e., different

parts of a function are performed by different subsystems in the MLDNE, while the final result of the functions remains transparent to the external nodes and end-stations. As will be appreciated from the discussion below and the diagram in Figure 1, the MLDNE has a scalable architecture which allows
5 the designer to predictably increase the number of external connections by adding additional subsystems, thereby allowing greater flexibility in defining the MLDNE as a stand alone router.

As illustrated in block diagram form in Figure 1, the MLDNE 101 contains a number of subsystems 110 that are fully meshed and
10 interconnected using a number of internal links 141 to create a larger switch. At least one internal link couples any two subsystems. Each subsystem 110 includes a switch element 111 coupled to a forwarding memory 113 and an associated memory 114. The forwarding memory (or database) 113 stores an address table used for matching with the headers of received packets. The
15 associated memory (or database) stores data associated with each entry in the forwarding memory that is used to identify forwarding attributes for forwarding the packets through the MLDNE. A number of external ports (not shown) having input and output capability interface the external connections 117. In one embodiment, each subsystem supports multiple Gigabit Ethernet
20 ports, Fast Ethernet ports and Ethernet ports. Internal ports (not shown) also having input and output capability in each subsystem couple the internal links 141. Using the internal links, the MLDNE can connect multiple switching elements together to form a multigigabit switch.

The MLDNE 101 further includes a central processing system (CPS) 160
25 that is coupled to the individual subsystem 110 through a communication bus 151 such as the peripheral components interconnect (PCI). The CPS 160 includes a central processing unit (CPU) 161 coupled to a central memory 163.

Central memory 163 includes a copy of the entries contained in the individual forwarding memories 113 of the various subsystems. The CPS has a direct control and communication interface to each subsystem 110 and provides some centralized communication and control between switch
5 elements.

Figure 2 is a simplified block diagram illustrating an exemplary architecture of the switch element of Figure 1. The switch element 200 depicted includes a central processing unit (CPU) interface 215, a switch fabric block 210, a network interface 205, a cascading interface 225, and a shared
10 memory manager 220.

Ethernet packets may enter or leave the network switch element 200 through any one of the three interfaces 205, 215, or 225. In brief, the network interface 205 operates in accordance with a corresponding Ethernet protocol to receive Ethernet packets from a network (not shown) and to transmit
15 Ethernet packets onto the network via one or more external ports (not shown). An optional cascading interface 225 may include one or more internal links (not shown) for interconnecting switching elements to create larger switches. For example, each switch element may be connected together with other switch elements in a full mesh topology to form a multi-layer
20 switch as described above. Alternatively, a switch may comprise a single switch element 200 with or without the cascading interface 225.

The CPU (not shown) may transmit commands or packets to the network switch element 200 via the CPU interface 215. In this manner, one or more software processes running on the CPU may manage entries in an
25 external forwarding and filtering database 240, such as adding new entries and invalidating unwanted entries. In alternative embodiments, however, the CPU may be provided with direct access to the forwarding and filtering

database. In any event, for purposes of packet forwarding, the CPU port of the CPU interface 215 resembles a generic input port into the switch element 200 and may be treated as if it were simply another external network interface port. However, since access to the CPU port occurs over a bus such as a peripheral components interconnect (PCI) bus, the CPU port does not need
5 any media access control (MAC) functionality.

Returning to the network interface 205, the two main tasks of input packet processing and output packet processing will now briefly be described. Input packet processing may be performed by one or more input ports of the network interface 205. Input packet processing includes the following: (1)
10 receiving and verifying incoming Ethernet packets, (2) modifying packet headers when appropriate, (3) requesting buffer pointers from the shared memory manager 220 for storage of incoming packets, (4) requesting forwarding decisions from the switch fabric block 210, (5) transferring the incoming packet data to the shared memory manager 220 for temporary
15 storage in an external shared memory 230, and (5) upon receipt of a forwarding decision, forwarding the buffer pointer(s) to the output port(s) indicated by the forwarding decision. Output packet processing may be performed by one or more output ports of the network interface 205. Output
20 processing includes requesting packet data from the shared memory manager 220, transmitting packets onto the network, and requesting deallocation of buffer(s) after packets have been transmitted.

The network interface 205, the CPU interface 215, and the cascading interface 225 are coupled to the shared memory manager 220 and the switch
25 fabric block 210. Preferably, critical functions such as packet forwarding and packet buffering are centralized as shown in Figure 2. The shared memory manager 220 provides an efficient centralized interface to the external shared

memory for buffering of incoming packets. The switch fabric block 210 includes a search engine and learning logic for searching and maintaining the forwarding and filtering database with the assistance of the CPU.

The centralized switch fabric block 210 includes a search engine that provides access to the forwarding and filtering database on behalf of the interfaces 205, 215, and 225. Packet header matching, Layer 2 based learning, Layer 2 and Layer 3 packet forwarding, filtering, and aging are exemplary functions that may be performed by the switch fabric block 210. Each input port is coupled with the switch fabric block 210 to receive forwarding decisions for received packets. The forwarding decision indicates the outbound port(s) (e.g., external network port or internal cascading port) upon which the corresponding packet should be transmitted. Additional information may also be included in the forwarding decision to support hardware routing such as a new MAC destination address (DA) for MAC DA replacement. Further, a priority indication may also be included in the forwarding decision to facilitate prioritization of packet traffic through the switch element 200.

In the present embodiment, Ethernet packets are centrally buffered and managed by the shared memory manager 220. The shared memory manager 220 interfaces every input port and output port and performs dynamic memory allocation and deallocation on their behalf, respectively. During input packet processing, one or more buffers are allocated in the external shared memory and an incoming packet is stored by the shared memory manager 220 responsive to commands received from the network interface 205, for example. Subsequently, during output packet processing, the shared memory manager 220 retrieves the packet from the external shared memory and deallocates buffers that are no longer in use. To assure no buffers are

released until all output ports have completed transmission of the data stored therein, the shared memory manager 220 preferably also tracks buffer ownership.

Figure 3 is a simplified block diagram of the structure for implementing high speed field replacement in accordance with the teachings of the present invention. The following elements are used: an input Media Access Control (MAC 305), input port process (IPP) 310, search engine 315, database 320, packet memory 325, output port process (OPP) 330 and output MAC 335. As is readily apparent from the prior overview of the switching element (Figure 2), many of the elements are multi-purpose and provide additional functionality. Thus the present structure is not only time efficient but also cost effective, adding a minimum amount of additional logic to the structure in order to support field replacement.

Furthermore, the present invention preserves end-to-end error robustness by modifying or updating header information only when necessary. For example, in prior art systems, whether implemented in hardware or software, the receive MAC is configured to always strip the CRC off the packet regardless of whether the header of the packet is modified. Thus, the transmit MAC in such prior art devices is configured to always generate the CRC. As will be explained below, the CRC is only stripped when the header is modified, if the header is not modified, the original CRC remains untouched and end-to-end error robustness is preserved.

For purposes of simplifying the present discussion, the additional functional details of the different elements not directly related to the field replacement process described therein are not discussed in detail. Furthermore, it is contemplated that the structure described herein can be applied to other switching elements having similar structures. Finally,

although field replacement for VLAN packets and hardware routing is described, it is contemplated that other types of packets similarly requiring header field replacement can benefit from the teachings of the present invention.

5 Referring to Figure 3, the input MAC 305 receives the input packet and directs the input packet to the IPP 310. The IPP includes a first in first out (FIFO) buffer 312 to buffer the input packet. Logic 314, which preferably includes multiplexing circuitry and the associative select logic, is also included to forward control information and also replace predetermined
10 fields of the header as the header is transmitted out of the IPP 310 and into the packet memory 325. The IPP 310 forwards a copy of the header to the search engine 315 which searches the database 320 to determine if there is information relevant to the packet such as the type of packet, e.g., VLAN supported or whether the packet can be routed. It is contemplated that a
15 variety of configurations of search engines and databases may be used. In one embodiment, the search engine 315, in conjunction with the database 320, determines whether the input packet can be routed in hardware. A variety of search criteria may be used to determine this, including whether a route already exists for the input packet. In another embodiment, the search
20 engine returns information regarding VLAN support.

 The search engine 315 returns the information to the IPP 310. If the information indicated that the header is not to be updated, the header is output via the packet memory 325, OPP 330, and MAC 335 unchanged. Otherwise, the IPP 310 outputs the header from the FIFO and selectively
25 performs on the fly field replacement for predetermined fields of the header. The search engine provides an offset which identifies the location of the time to live (TTL) field in the packet. In addition, for unicast routing, the IPP 310

replaces the destination address (DA) field with a DA supplied by the search engine 315. For example, this can simply be done using multiplexor logic to select either the original value found in the DA field or the DA value received from the search engine 315 based upon a DA replacement control
5 signal issued by the search engine 315. For VLAN support, the IPP 310 selectively replaces or inserts the value in the VLAN tag field.

A counter is preferably used in the IPP 310 to count the bytes output and thus determine which field of the packet is currently being output such that the replacement is timely performed. In addition, it is preferred that the
10 IPP 310 adjusts the time to live (TTL) value in the TTL field and the checksum value in the checksum field of the header. The location of the TTL field is identified by the offset provided by the search engine. The checksum field is immediately following the TTL field. The TTL value output to the packet memory 325 is the TTL value found in the input packet header
15 decremented by one. Similarly, the checksum output is the checksum value from the input packet header decremented by a constant (due to the decrement of the TTL value).

In addition to selectively performing header field replacement, the IPP 310 outputs control field information which is stored in a control field 327 in
20 the packet memory 325. For example, for hardware routing, the control field information consists of an indication (replace_sa) to replace the source address. For VLAN support, the control field information consists of indicators orig_tag, mod_tag and dont_tag. The orig_tag indicator indicates that the packet originally arrived tagged. The mod_tag indicator indicates
25 that the packet originally arrived tagged but it is to be modified. The dont_tag indicator indicates that the packet is not to be tagged.

The packet memory 325 receives the modified input packet from the IPP 310 and the associative field control information. The packet memory functions as a buffer to minimize dropped packets during the movement of packets into and out of the switch.

5 The OPP 330, retrieves the packet and associated control field information from the packet memory 325 and, in response to the associated control field information, selectively modifies the input packet further as it is output to the MAC 335 and provides control information to the output MAC 335. For example, for hardware routing, the OPP 330 strips the last 4 bytes of
10 the packet containing the CRC and sends control signals to the MAC 335 to insert its address in the SA field and generate a CRC. In one embodiment, the OPP 330 provides instruction to the output MAC 335 by clearing the NO_CRC bit in the MAC control word subsequently sent to the MAC 335 to tell the MAC 335 to append a CRC. CRC generation and insertion is a typical
15 function found in MACs as well as other devices and will not be discussed in detail herein. Furthermore, in one embodiment the OPP 330 further issues a control signal to the MAC 335 to notify the MAC 335 to replace the SA value with its address. Similarly, for VLAN support, the OPP 330 selectively strips the value in the VLAN tag field, selectively strips the CRC from the packet
20 and clears the No_CRC bit in the MAC control word to notify the MAC 335 to append a CRC.

 The output MAC 335, responsive to the state of the control signals received, selectively generates a CRC and, inserts its own address in the source address field, completing the header field replacement process. The
25 modified packet is then output from the switch element.

 Figures 4a and 4b are simplified diagrams of two illustrative packet formats that are modified using the system described. Figure 4a shows a

packet consisting of data 402, the Layer 4 header (TCP header) 404, Layer 3 header (IP header) 406, Layer 2 header (data routing or MAC header) 408 and CRC 410. The Layer 2 header includes the DA field, 412, SA field 414, and packet type/length field 416. The Layer 3 header includes the time to live
5 field 418 and the checksum field 420. Figure 4b illustrates a VLAN supported packet. In this format the Layer 2 header, 408 is modified to include additional 4 bytes 422 that form the VLAN tag.

The process performed will now be generally described with reference to Figure 5. As described earlier, the process is applicable to field replacement,
10 including field replacement required for hardware routing and VLAN support. At step 510, the IPP receives an input packet and buffers the packet in the IPP FIFO. A copy of the header is forwarded to the search engine. At step 515, the search engine searches the database and determines the type of the packet. The type information and certain field replacement values are
15 returned to the IPP. At step 517, if the information supplied by the search engine indicates that the header is not to be modified, the packet is output unchanged from the switch element. Thus, end-to-end error robustness is maintained as the CRC is regenerated only when the header is changed. If header field replacement is needed, at step 520, the IPP selectively performs
20 initial field replacement as the data is transmitted from the FIFO to the packet memory. The fields replaced are replaced with values provided by the search engine and those computed (e.g., TTL, Checksum) in accordance with known techniques. In addition, the IPP forwards certain control field information to be stored in a control field associated with the particular location in the packet
25 memory the modified input packet is stored in.

At step 525, the OPP accesses the packet memory and associated control field information, selectively further modifies the packet and forwards the

packet and control information to the MAC. The MAC, at step 530, selectively modifies certain fields of the packet as the packet is output from the switch element.

Figure 6 illustrates one embodiment of the process that performs header field replacement for input packets that are hardware routed. At step 5605, the input packet is received by the IPP and the packet header is stored in the IPP FIFO. As the header is stored in the FIFO, step 610, a copy of the header is forwarded to the search engine, step 615, which searches the database to determine if the packet is to be routed. The IPP then waits for the search engine to return information regarding the packet, step 620. If the search engine determines that the input packet is a unicast route, step 625, the search engine, sends a replace_DA (destination address) control signal and replace_SA (source address) control signal to the IPP, provides the replacement DA and further provides a time to live (TTL) field offset. The IPP, in response to the replace_DA control signal, replaces the DA field value with the value received from the search engine computes an updated TTL value and checksum value and replaces the computed values in the corresponding fields as the fields are output to the packet memory. The IPP further responds to the replace_SA signal by inputting corresponding control field information in the packet memory to indicate that the source address is to be replaced with the source address of the output MAC. If at step 645, the search engine indicates that it is a multicast route, the search engine provides the TTL offset and sends a replace_SA signal to the IPP, step 650, and the IPP updates the TTL and checksum values and generates control field information that is stored in the control field associated with the packet memory, step 640.

At step 655, the modified packet output and associated control field information is stored in the packet memory. At step 660, the OPP receives data and control field information from the packet memory and if the OPP detects the replace_SA control field, step 665, the OPP asserts a replace_SA control signal to the output MAC. At step 670, the OPP strips the last 4 bytes of the packet corresponding to the CRC and clears the NO_CRC bit in the MAC control word. At step 680, the MAC detects the replace_SA control signal and replaces bytes 7-12 of the packet with its own MAC address during output packet transmission. Furthermore, in response to the state of the control word, the MAC generates the CRC for the packet. If at step 665, the replace_SA control field information does not indicate replacement, the MAC transmits the packet unmodified, step 675.

Figures 7a and 7b illustrate the process for VLAN support. At steps 705, 710, 715 the IPP receives the input packet, buffers the packet and forwards the header to the search engine. The search engine determines and returns information regarding tagging to the waiting IPP, step 720, regarding whether the packet is tagged, the nature of the tagging and how to tag the packet before it is output from the switch element. A number of possible scenarios exist, including that the packet arrived untagged, the packet arrived tagged valid and the packet arrived tagged invalid. The invalid tag, in certain circumstances, is used to convey only priority information for the packet, rather than VLAN grouping of end nodes.

If, at step 725 the packet is tagged, and the packet is tagged invalid, step 730, control signals are sent to the IPP from the search engine that the insert_tag indicator is to be cleared indicating that no new tag is to be inserted and the replace_tag indicator is to be set, indicating that the tag is to be replaced, step 735. If, at step 730, the packet is tagged valid, and VLAN

routing is supported, step 740, at step 745, the insert_tag indicator is cleared, the replace_tag indicator is set and a new VLAN tag representative of the VLAN routing determined by the search engine is provided to the IPP. If at step 740, VLAN routing is not supported, at step 750, signals are sent to the
5 IPP to indicate that the insert_tag is cleared and replace_tag is cleared indicating that a tag is not to be inserted or replaced.

Returning back to step 725, if the packet arrives untagged, at step 755, signals are issued to the IPP indicating that the insert_tag indicator is to be set and the replace_tag indicator is to be cleared. Following the process flow, for
10 the scenarios of untagged packets and packets that are tagged invalid, it is determined whether the tag to be provided by the search engine is one that is defined in the database, step 760. If the tag is defined in the database, the tag is provided to the IPP, step 765. If the tag is not defined in the database, a default tag is provided, step 770. The default tag is a programmable value; typical
15 values follow those specified in current standards.

The IPP, in response to the state of the insert_tag, replace_tag and the VLAN tag value selectively provided by the search engine, will selectively modify the packet header and generate control field information; the modified packet and associated control field information are then respectively
20 stored in the packet memory and associated control fields. At step 772, if the insert_tag indicator was set, at step 774, the following control field information is generated: clear both orig_tag and mod_tag. Orig_tag indicates that the packet has arrived as tagged. Mod_tag indicates that the packet arrived tagged but the tag has been modified. Furthermore, at step 774,
25 the tag provided by the search engine is inserted at the appropriate place in the header by the IPP, preferably as the packet is output to be stored in the packet memory.

Returning back to step 772, if the insert_tag indicator is not set and the replace_tag indicator is set, step 776, at step 778, the IPP replaces the tag in the header with the tag provided by the search engine and generates the following control field information: set orig_tag and set mod_tag, step 780. If
5 at step 776 the replace_tag indicator is not set, the IPP generates the following control field information: set orig_tag, clear mod_tag, at next step 780.

In the present embodiment, the CPU of the network switch can communicate packets through the switch elements. If the packet arrived through this port, the packet may not be tagged regardless of the state of the
10 packet. Thus the indicator dont_tag is provided as control field information to the packet. Referring back to step 782, if the packet arrives through the host transmit process (HTP), dont_tag is set to equal the packet control information provided by the CPU; otherwise, at step 786, dont_tag is cleared.

At step 790, the packet is stored in the packet memory and the control
15 field information is stored in the associated control field. At step 792, the OPP retrieves the packet and control field information and decodes the control field information, step 794. The OPP decodes the three indicators retrieved from the packet memory, orig_tag, mod_tag and dont_tag and a fourth indicator, tag_enable. Tag_enable is an internal variable which indicates that
20 the device that is going to receive the packet to be output does not support VLAN routing. This variable is determined by a network management mechanism based on the underlying network topology. For example, if the receiving node does not support VLAN routing, the tag_enable bit will be cleared. The result of the decoding process indicates whether the OPP is to
25 strip the tag and whether the MAC is to generate a CRC. The OPP decodes according the following table:

	dont_tag	tag_enable	orig_tag	mod_tag	strip tag	regenerate CRC
	1	x	0	x	Y	N
	1	x	1	x	Y	N
	0	0	0	x	Y	N
5	0	0	1	x	Y	Y
	0	1	0	x	N	Y
	0	1	1	0	N	N
	0	1	1	1	N	Y

10 Thus at step 796, if the tag is to be stripped, the OPP removes the tag, preferably as the tag is transferred to the MAC, step 798. At step 800, if no CRC is to be generated, the OPP sends a signal indicating that no CRC is to be generated (e.g., set no_CRC), step 802, and the MAC transmits the packet as it is received. If the CRC is to be generated, at step 806, the last 4 bytes are
15 removed from the packet by the OPP, a signal to generate the CRC is sent to the MAC, (clear no-CRC), step 808, and at step 810, the MAC transmits the packet and generates the CRC to append to the end of the packet.

20 The process for field replacement has been described. Other variations are also contemplated. For example, the present switch consists of multiple switch elements wherein packets can be transferred between switch elements. When packets are transferred between switch elements, certain fields are selectively modified.

25 The invention has been described in conjunction with the preferred embodiment. It is evident that numerous alternatives, modifications, variations and uses will be apparent to those skilled in the art in light of the foregoing description.

CLAIMS

What is claimed is:

- 1 1. In a packet switch, an apparatus for selective header field
2 replacement comprising:
3 an input port process (IPP) coupled to receive the input packet
4 comprising a header, data and cycle redundancy code (CRC), said IPP further
5 comprising a buffer configured to temporarily store the input packet;
6 a database configured to store information regarding packets and
7 routes;
8 a search engine coupled between the IPP and the database, said search
9 engine coupled to receive the header and configured to search the database to
10 determine information regarding a type of the input packet ;
11 said IPP further configured to output the input packet from the buffer
12 and to selectively replace at least one field in the header in response to the
13 information provided by the search engine and selectively output control
14 field information to indicate additional modification of the modified input
15 packet prior to output from the switch;
16 an output port process (OPP) configured to receive the selectively
17 modified input packet and the control field information, said OPP configured
18 to selectively generate at least one control signal to notify that the modified
19 input packet is to be further modified prior to output from the switch and to
20 output the selectively modified input packet;
21 an output interface, said output interface coupled to receive the at least
22 one control signal and the selectively modified input packet and is configured
23 to output a packet from the switch that corresponds to the selectively
24 modified input packet, said output interface further configured to selectively

25 modify, in response to the at least one control signal, at least one header field
26 and the CRC prior to transmission of the output packet onto the medium.

1 2. The apparatus as set forth in claim 1, wherein the type comprises
2 an indication of whether the input packet is to be routed, wherein if the input
3 packet is to be routed, said search engine is configured to notify the IPP that
4 the input packet is to be routed and the destination address (DA) of the route.

1 3. The apparatus as set forth in claim 2, wherein said IPP is
2 configured to replace a DA field of the header with a DA provided by the
3 search engine if the search engine notifies the IPP that the input packet is to
4 be routed.

1 4. The apparatus as set forth in claim 2, wherein , said control field
2 information comprises a field set by the IPP to indicate that the source address
3 field of the header is to be replaced prior to output of the modified input
4 packet, said field set when the input packet is to be routed.

1 5. The apparatus as set forth in claim 2, wherein the at least one
2 control signal comprises control signals to selectively indicate generation of a
3 CRC and a replacement of a source address.

1 6. The apparatus as set forth in claim 5, wherein the output
2 interface is configured to insert the address of the output interface in a source
3 address field of the header in response to the receipt of the at least one control
4 signal indicating replacement of the source address, and to generate a CRC in
5 response to the at least one control signal indicating regeneration of the CRC.

1 7. The apparatus as set forth in claim 1, wherein the header
2 comprises a time to live (TTL) field, said IPP further configured to decrement
3 a value in the TTL field by one prior to output.

1 8. The apparatus as set forth in claim 1, wherein the header
2 comprises a checksum field, said IPP further configured to update a value in
3 the checksum field prior to output.

1 9. The apparatus as set forth in claim 2, wherein the destination
2 address is replaced when the packet is a unicast packet.

1 10. The apparatus as set forth in claim 5, wherein the OPP is further
2 configured to strip off the CRC during transmission of the modified input
3 packet to the output interface if the output interface is to generate the CRC.

1 11. The apparatus as set forth in claim 1, wherein the output
2 interface is a MAC.

1 12. The apparatus as set forth in claim 11, wherein the at least one
2 control signal comprises a replace_sa signal.

1 13. The apparatus as set forth in claim 11, wherein the at least one
2 control signal comprises a state of a NO_CRC bit in a control word
3 transmitted to the MAC by the OPP.

1 14. The apparatus as set forth in claim 1, wherein the switch
2 supports virtual local area networks (VLANs) and the type comprises an
3 indication of whether the input packet untagged, tagged with a valid tag or
4 tagged with an invalid tag, wherein:
5 if the input packet is untagged, and the search engine determines that
6 the input packet belongs to a VLAN that has been defined, said search engine
7 is configured to notify the IPP of a VLAN tag of the defined VLAN, and at
8 least one indicator indicating that the VLAN tag is to be inserted into a VLAN
9 tag field of the header;
10 if the input packet is tagged with an invalid tag, and the search engine
11 determines that the input packet belongs to a VLAN that has been defined,
12 said search engine is configured to notify the IPP of the VLAN tag of the
13 defined VLAN, and at least one indicator indicating that the VLAN tag is to
14 be replaced;
15 if the input packet is tagged with an valid tag, and the search engine
16 determines that the input packet should be routed to a different VLAN, said
17 search engine is configured to notify the IPP of the VLAN tag of the different
18 VLAN, and at least one indicator indicating that the VLAN tag is to be
19 replaced; and
20 otherwise the search engine is configured to notify the IPP of at least
21 one indicator that indicates no insertion or replacement of the VLAN tag.

1 15. The apparatus as set forth in claim 14, wherein the IPP is
2 configured to:
3 if the VLAN tag is to be inserted, insert the VLAN tag provided by the
4 search engine, and generate control field information of a first state;

5 if the VLAN tag is to be replaced, replace the VLAN tag in the input
6 packet header with the VLAN tag provided by the search engine and generate
7 control field information of a second state;
8 otherwise, generate control field information of a third state.

1 16. The apparatus as set forth in claim 15, wherein said OPP is
2 further configured to selectively remove the VLAN tag field in the modified
3 input packet based on the state of the control field information.

1 17. The apparatus as set forth in claim 15, wherein said OPP is
2 configured to selectively generate a control signal to regenerate a CRC based
3 on the state of the control field information.

1 18. The apparatus as set forth in claim 17, wherein the output
2 interface is configured to generate a CRC in response to the at least one
3 control signal indicating regeneration of the CRC.

1 19. The apparatus as set forth in claim 1, further comprising a packet
2 memory configured to store a selectively modified input packet received from
3 the IPP, said packet memory further configured to store control field
4 information, said OPP coupled to the packet memory and configured to
5 receive the selectively modified input packet and the control field
6 information from the packet memory.

1 20. A method for selectively performing header field replacement in
2 a network switch device comprising the steps of:
3 an input port process (IPP) buffering an input packet;

4 a search engine searching a database to determine a type of the input
5 packet and notifying the IPP of the type of the input packet;
6 said IPP selectively replacing at least one field in the header in response
7 to the information provided by the search engine and selectively outputting
8 control field information to indicate additional modification of the modified
9 input packet prior to output from the switch;
10 an output port process (OPP) reading a modified input packet and
11 corresponding control field information, said OPP selectively generating at
12 least one control signal to an output interface to notify that the modified
13 input packet is to be further modified prior to output from the switch and to
14 output the selectively modified input packet;
15 said output interface outputting the modified input packet further
16 selectively modified in response to the at least one control signal received
17 from the OPP.

1 21. The method as set forth in claim 20, further comprising the step
2 of storing the modified input packet and control field information in a packet
3 memory; said OPP further reading the modified input packet and control field
4 information from the packet memory.

1 22. The method as set forth in claim 20, wherein the type comprises
2 an indicating of whether the input packet is to be routed, wherein if the input
3 packet is to be routed, said search engine notifying the IPP that the input
4 packet is to be routed and the destination address (DA) of the route.

1 23. The method as set forth in claim 22, wherein the step of the IPP
2 selectively replacing comprises the step of replacing a DA field of the header
3 with a DA provided by the search engine if the input packet is to be routed.

1 24. The method as set forth in claim 22, wherein , said control field
2 information comprises a field (replace_SA) set by the IPP to indicate that the
3 source address field of the header is to be replaced prior to output of the
4 modified input packet, said replace_SA set when the input packet is to be
5 routed.

1 25. The method as set forth in claim 22, wherein the step of
2 generating at least one control signal comprises the step of generating control
3 signals to selectively indicate generation of a CRC and a replacement of a
4 source address.

1 26. The method as set forth in claim 25, wherein the step of
2 outputting comprises the steps of:
3 inserting the address of the output interface in the source address field
4 of the header in response to the receipt of the at least one control signal
5 indicating replacement of the source address; and
6 generating a CRC in response to the at least one control signal
7 indicating regeneration of the CRC.

1 27. The method as set forth in claim 20, wherein the switch supports
2 virtual local area networks (VLANs) and the type comprises an indication of
3 whether the input packet untagged, tagged with a valid tag or tagged with an
4 invalid tag, wherein:

5 if the input packet is untagged, and the search engine determines that
6 the input packet belongs to a VLAN that has been defined, said search engine
7 notifying the IPP of a VLAN tag of the defined VLAN, and further issuing at
8 least one indicator indicating that the VLAN tag is to be inserted into a VLAN
9 tag field of the header;
10 if the input packet is tagged with an invalid tag, and the search engine
11 determines that the input packet belongs to a VLAN that has been defined,
12 said search engine notifying the IPP of the VLAN tag of the defined VLAN,
13 and further issuing at least one indicator indicating that the VLAN tag is to be
14 replaced;
15 if the input packet is tagged with a valid tag, and the search engine
16 determines that the input packet should be routed to a different VLAN, said
17 search engine notifying the IPP of the VLAN tag of the different VLAN, and
18 issuing at least one indicator indicating that the VLAN tag is to be replaced;
19 and
20 otherwise the search engine notifying the IPP of at least one indicator
21 that indicates no insertion or replacement of the VLAN tag.

1 28. The method as set forth in claim 27, wherein:

2 if the VLAN tag is to be inserted, said step of said IPP selectively
3 replacing comprising the step of inserting the VLAN tag provided by the
4 search engine, and said step of selectively outputting control field
5 information comprising the step of generating control field information of a
6 first state;

7 if the VLAN tag is to be replaced, said step of said IPP selectively
8 replacing comprising the step of replacing the VLAN tag in the input packet
9 header with the VLAN tag provided by the search engine and said step of

10 selectively outputting control field information comprising the step of
11 generating control field information of a second state;
12 otherwise, said step of selectively outputting control field information
13 comprising the step of generating control field information of a third state.

1 29. The method as set forth in claim 28, further comprising the step
2 of said OPP selectively removing the VLAN tag field in the modified input
3 packet based on the state of the control field information.

1 30. The method as set forth in claim 28, wherein the step of
2 selectively generating at least one control signal comprises a control signal to
3 regenerate a CRC based on the state of the control field information.

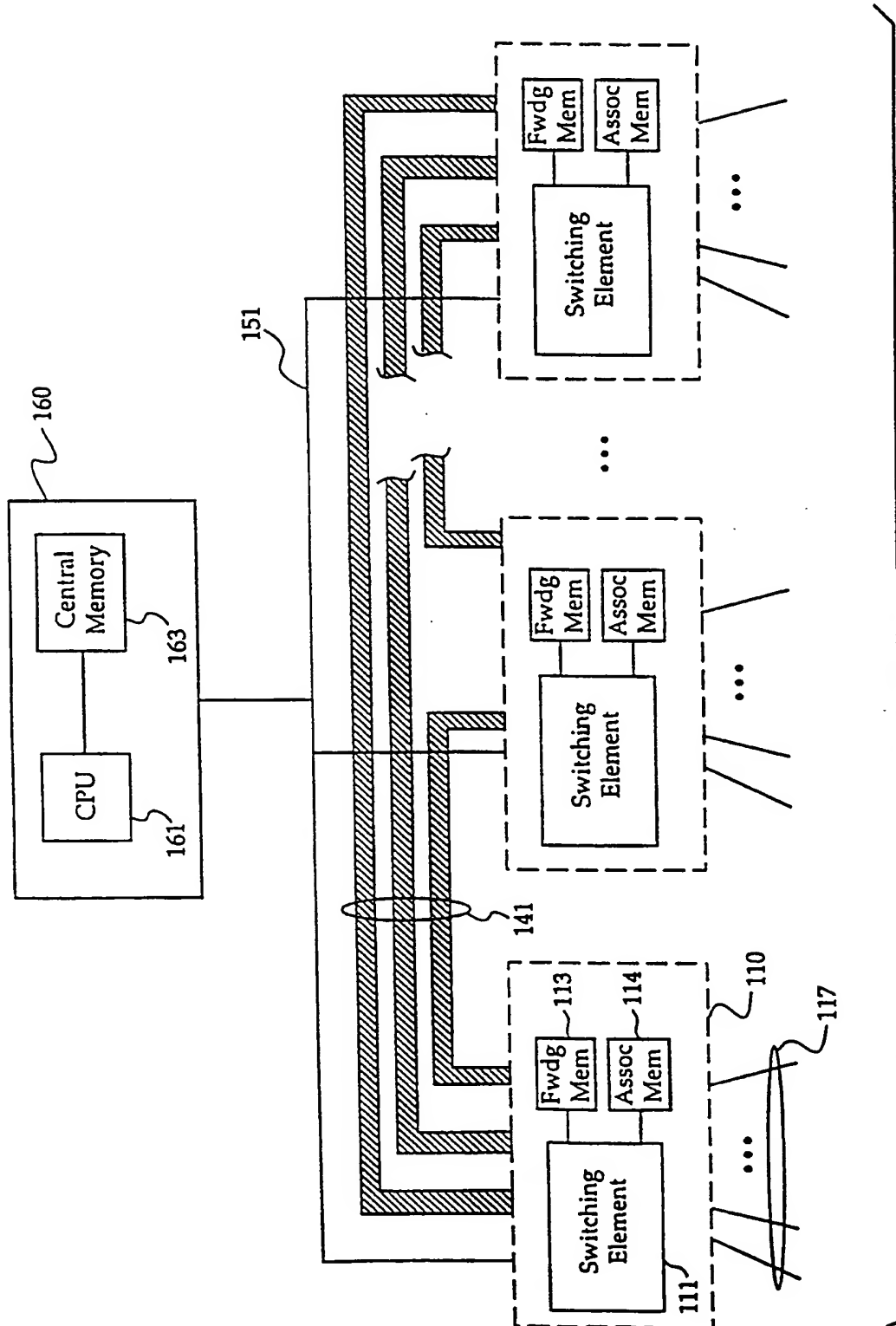
1 31. The method as set forth in claim 30, wherein the output
2 interface further selectively modifies the modified input packet by generating
3 a CRC in response to the at least one control signal indicating regeneration of
4 the CRC.

1 32. In a packet switch, an apparatus for selective header field
2 replacement comprising:
3 a switch element coupled to receive header information of an input
4 packet comprising a header, data and cycle redundancy code (CRC), said
5 switch element configured to determine information regarding a type of the
6 input packet;
7 if the information determined by the switch element indicates that the
8 header does not change, the input packet is output from the switch without
9 modification thereby preserving end-to-end error robustness;

10 if the information determined by the switch element indicates that at
11 least one portion of the header changes, the header of the input packet is
12 updated, the CRC is stripped and regenerated based upon the updated packet
13 wherein an updated packet is output from the switch.

1 33. The apparatus as set forth in claim 32, wherein said switch
2 element comprises:
3 a database configured to store information regarding packets;
4 a search engine coupled to the database and coupled to receive the
5 header information of the input packet, said search engine configured to
6 search the database to determine information regarding a type of the input
7 packet , wherein the search engine determines whether at least one portion of
8 the header changes.

1 34. The apparatus as set forth in claim 32, wherein the switch
2 element further comprising logic that receives information regarding a node
3 subsequently receiving the packet to be output from the switch, said logic
4 determining whether at least one portion of the header changes.



To nodes and end-stations

FIG. 1

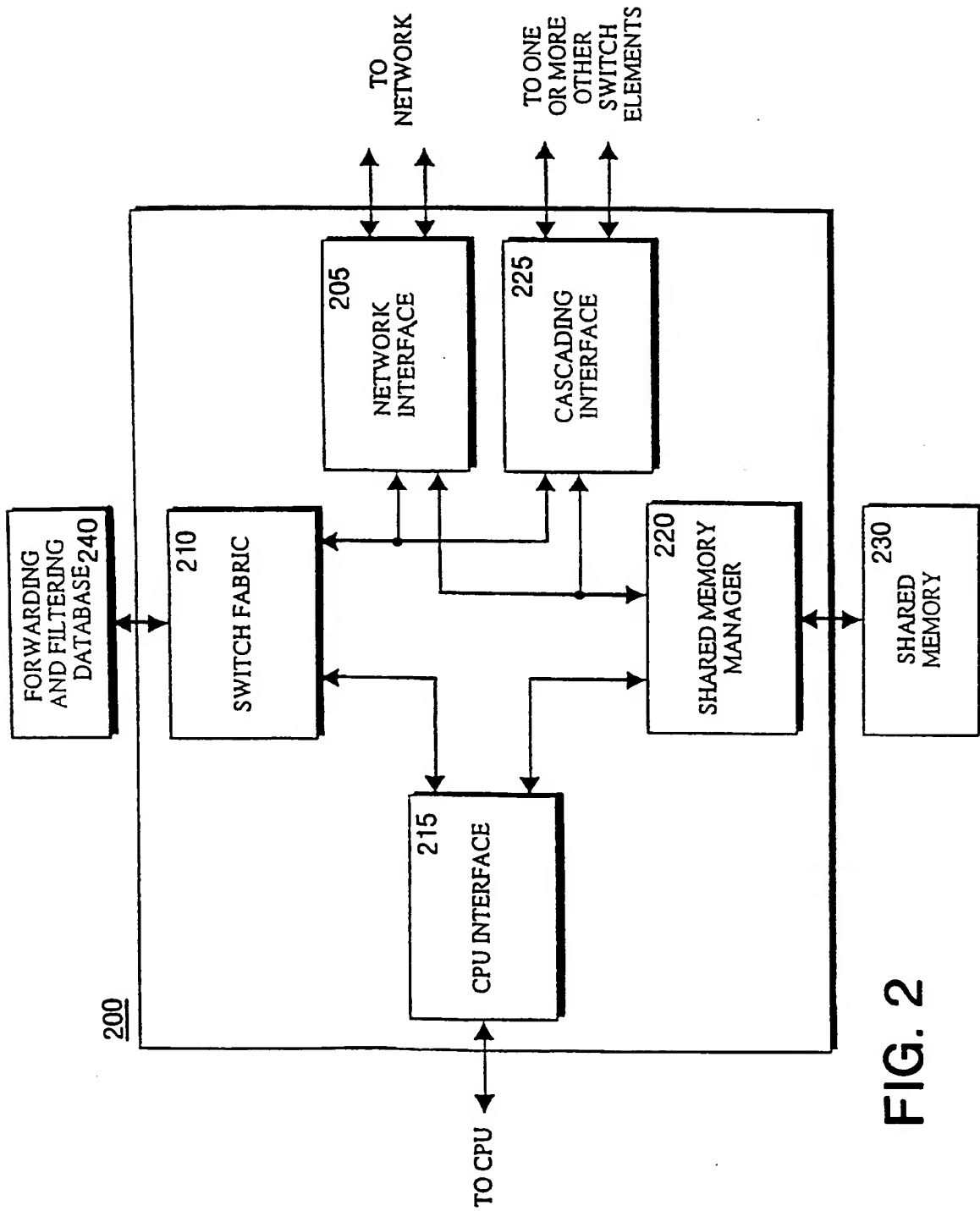


FIG. 2

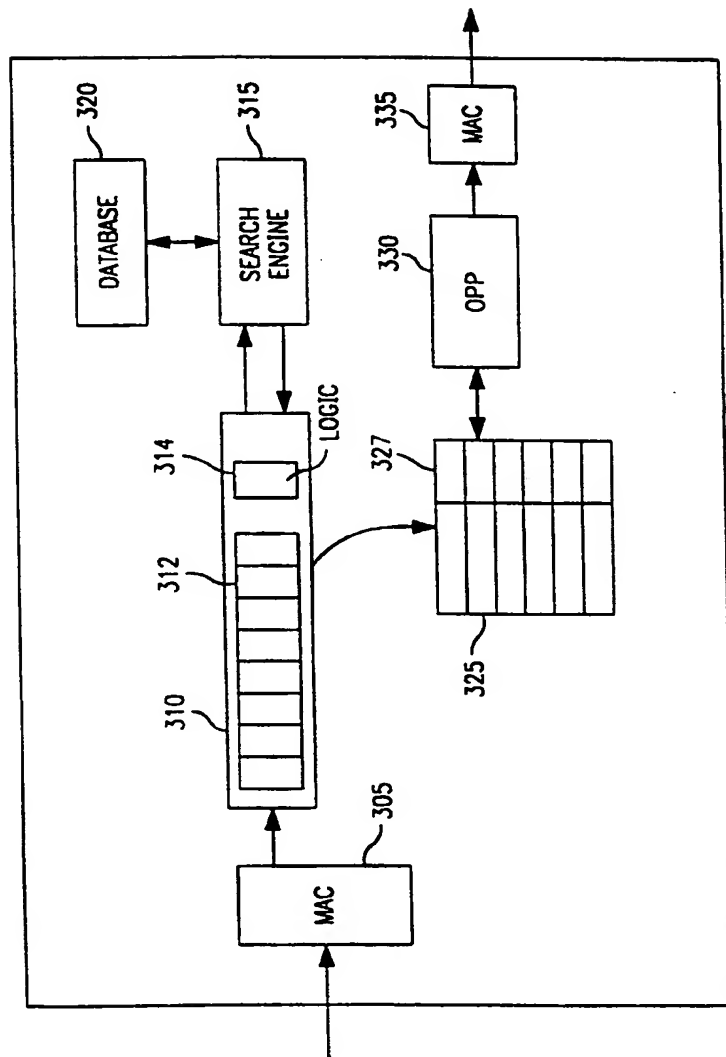


FIG. 3

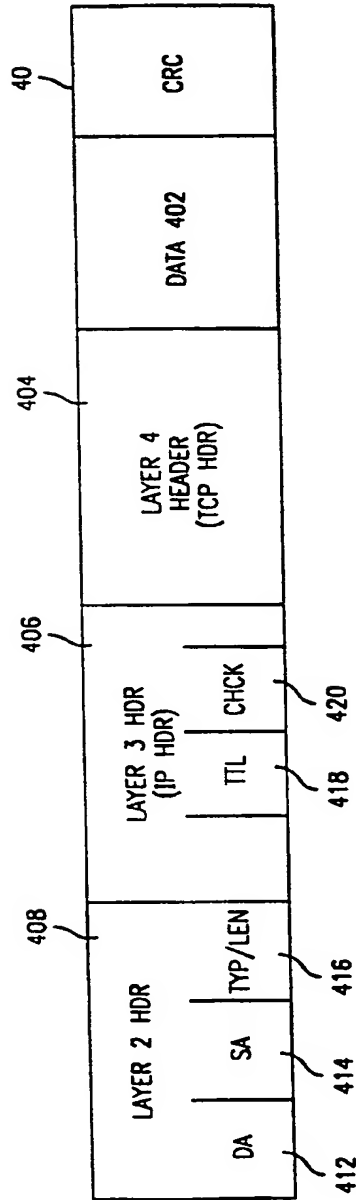


FIG. 4A

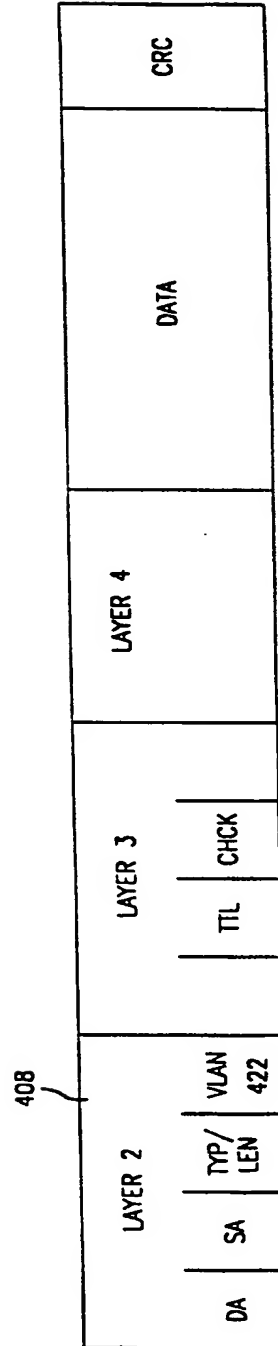


FIG. 4B

5/8

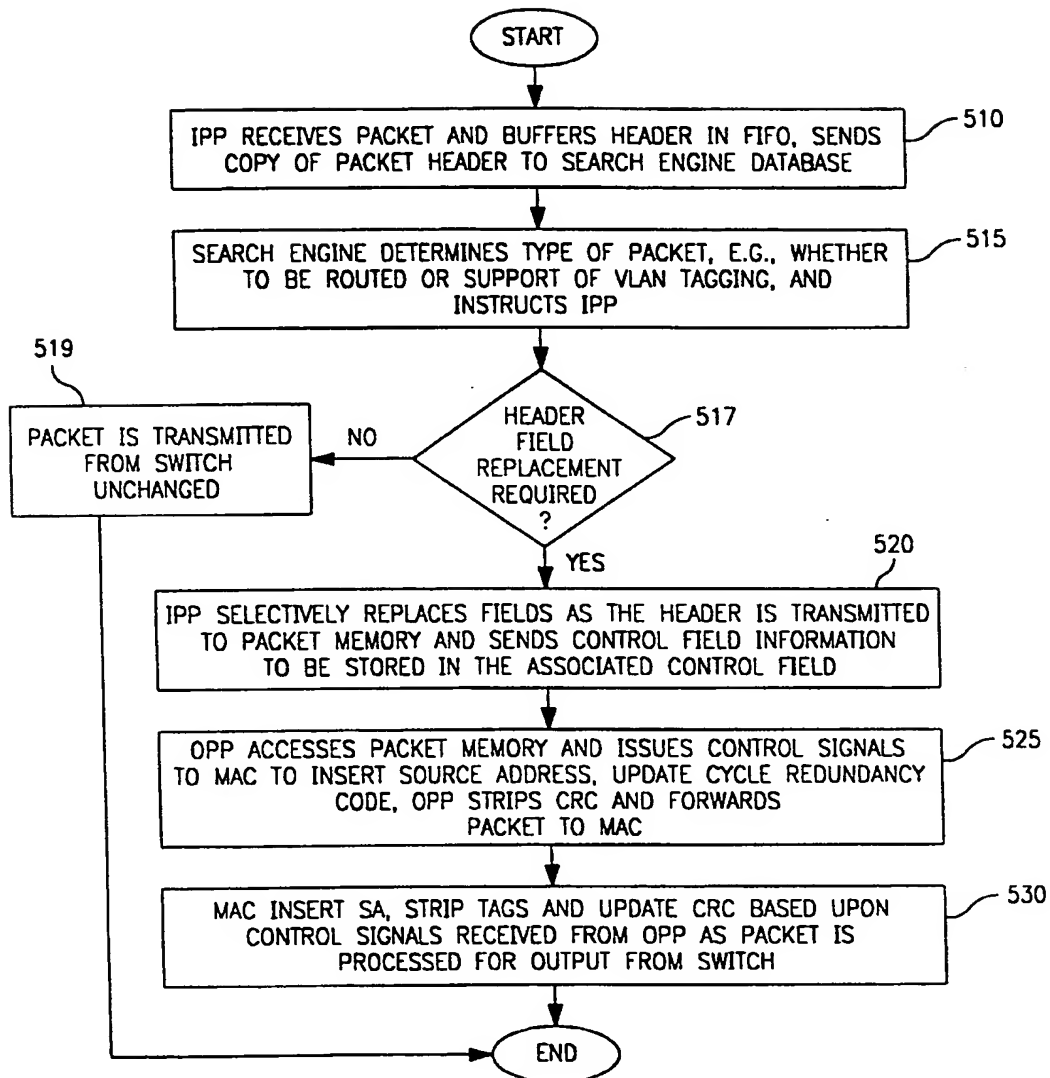


FIG. 5

6/8

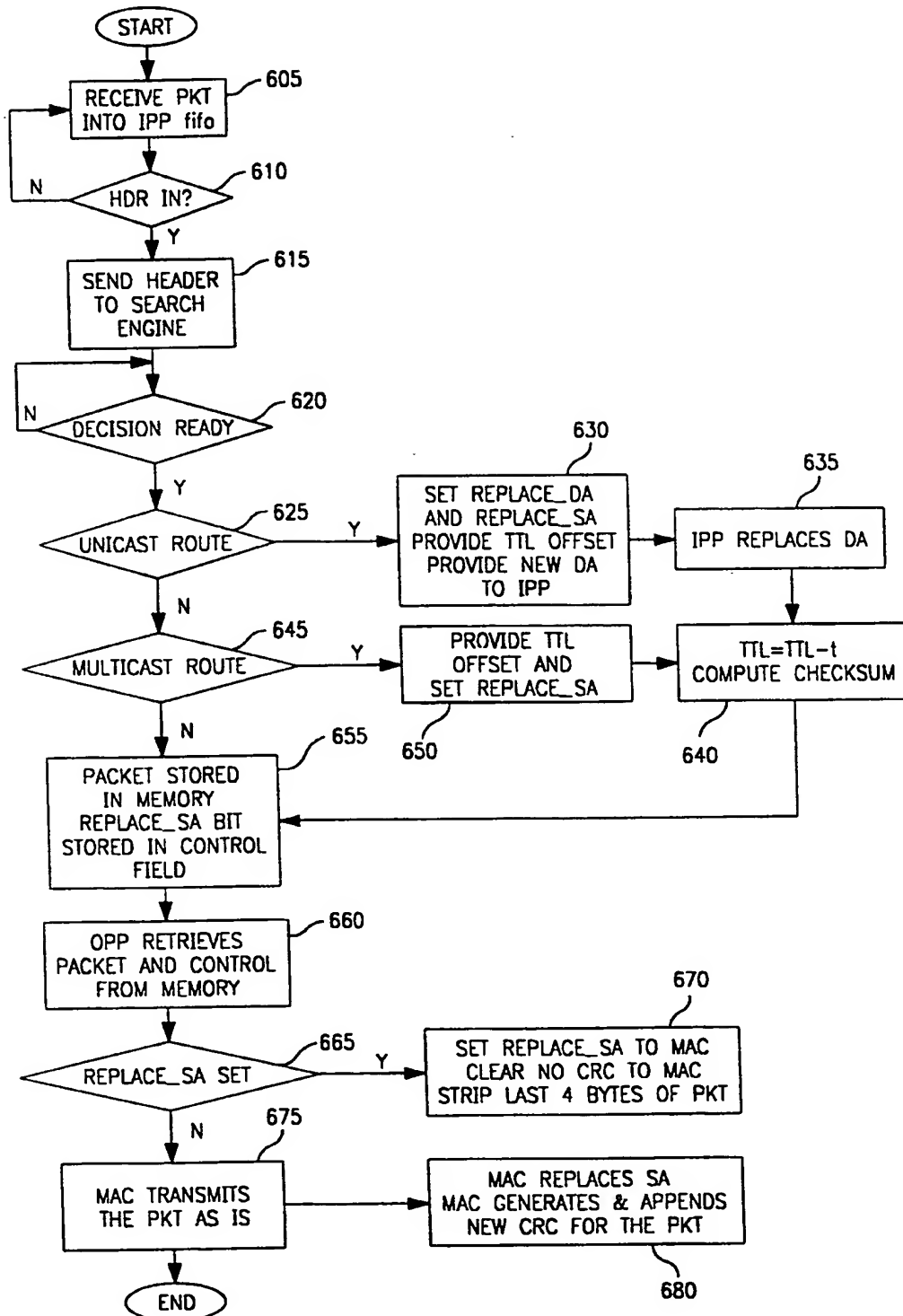
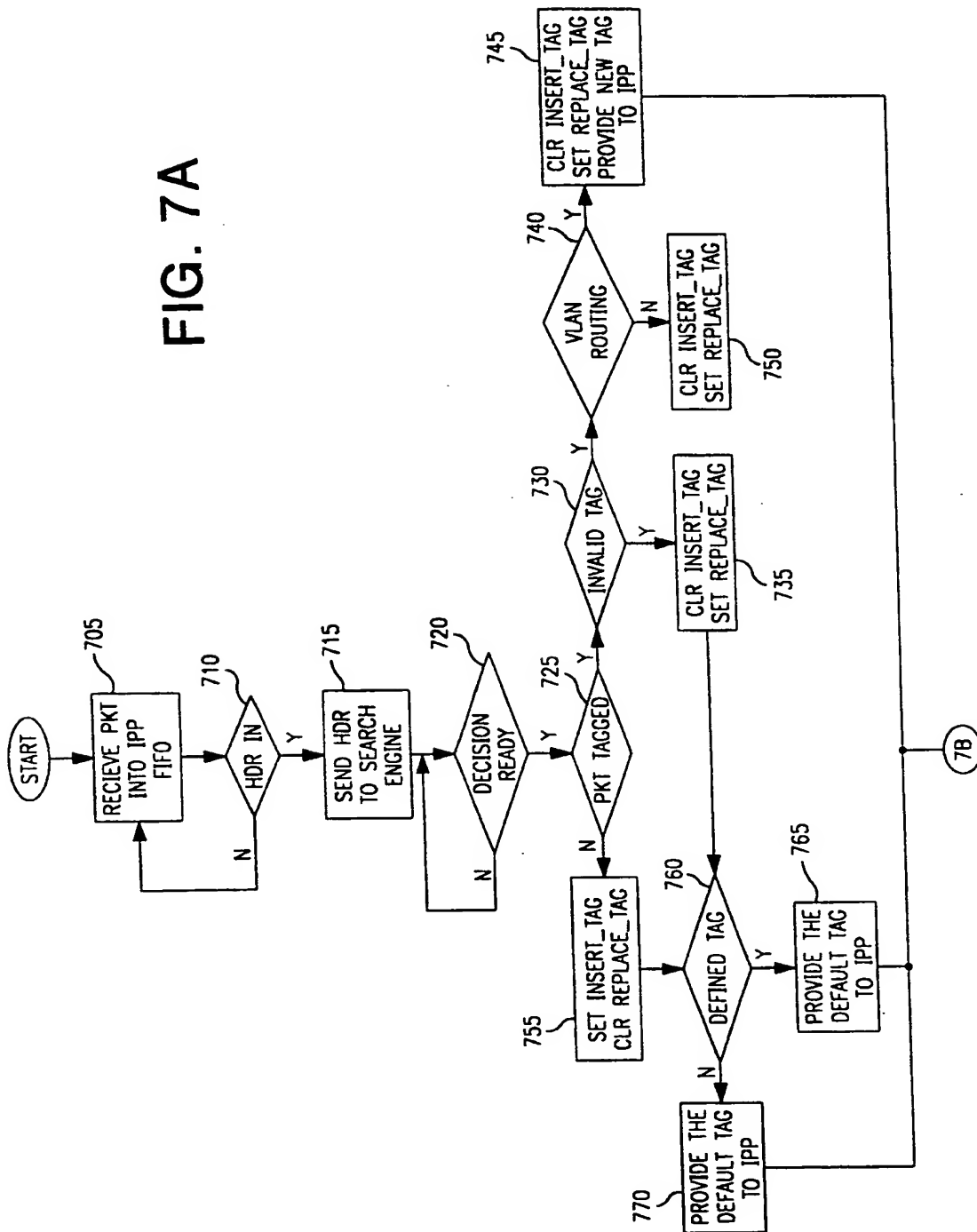


FIG. 6

FIG. 7A



8/8

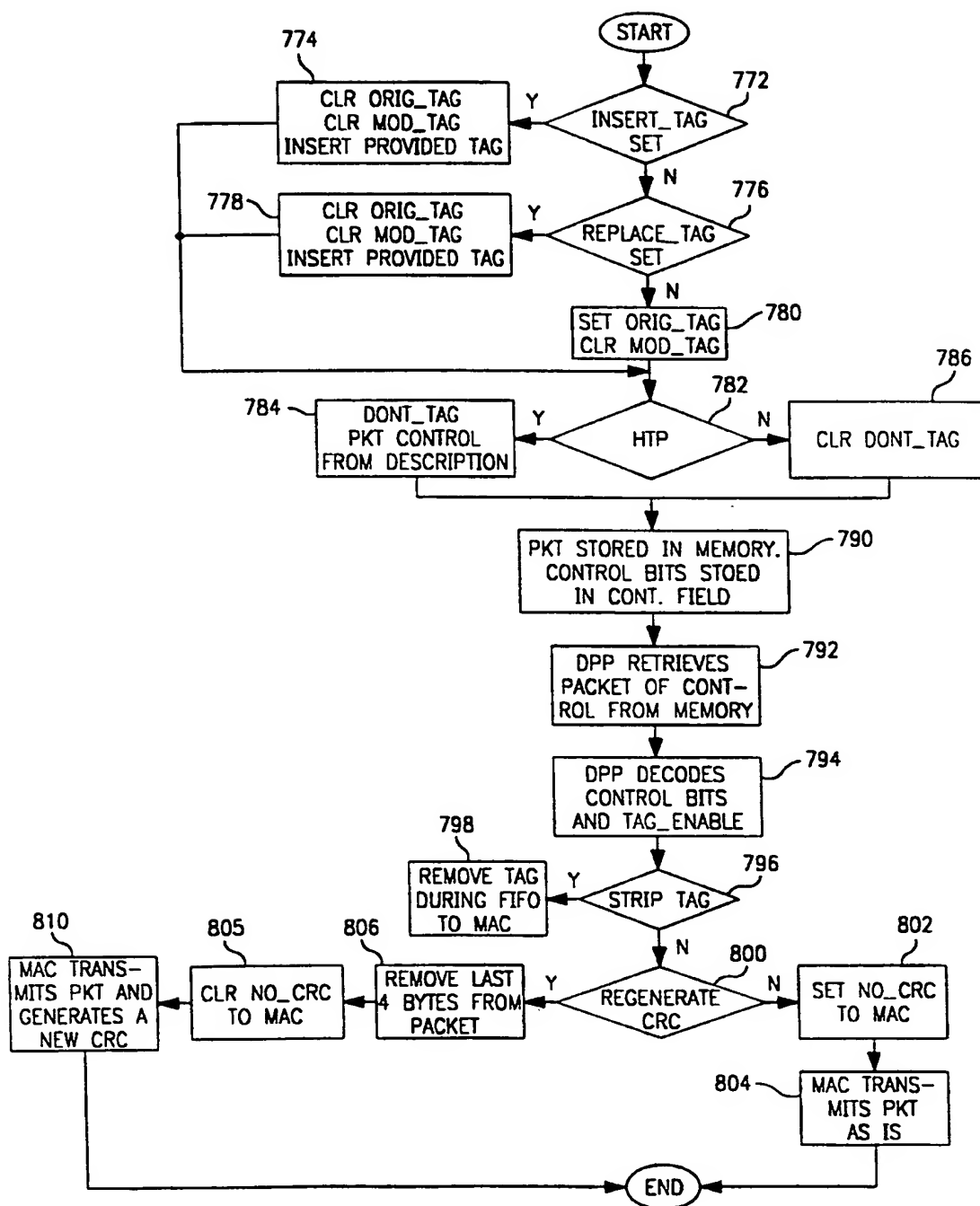


FIG. 7B